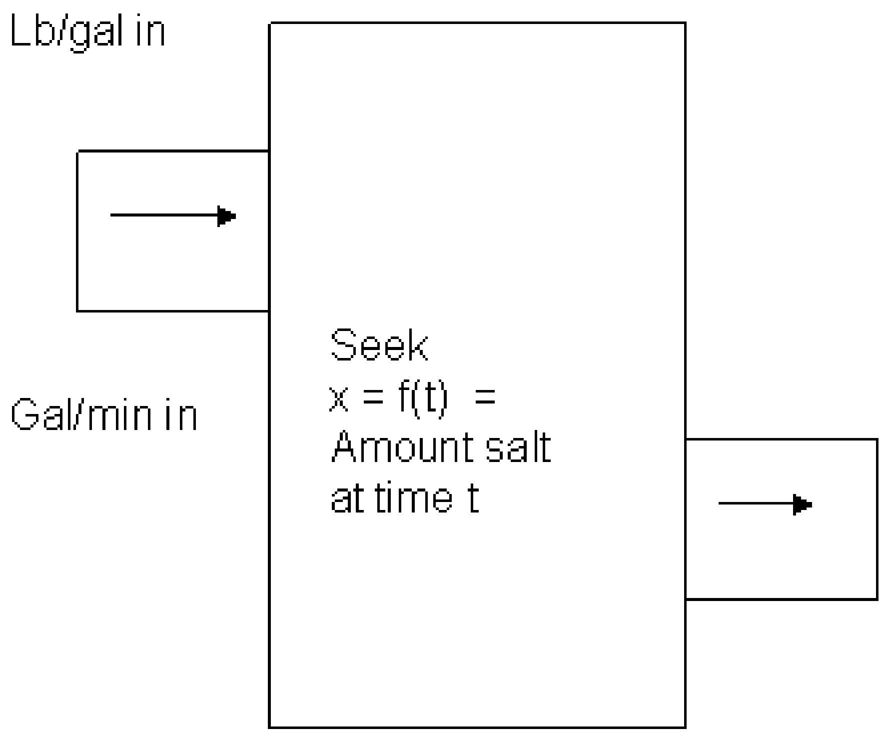
Mixture Problem from Differential Equations

Mixture Problem Discussion. Can open Output section and proceed if you are in a hurry.

$\text{Ode} = \frac{d}{dt} x(t) = \text{In} - \text{Out} =$

LbPerGalIn*galPerMinIn -galPerMinOut*x(t)/(VolOfH20AtStart+(galPerMinIn-galPerMinOut)*t)



Code for solving ode and displaying results. Module keeps track of various variables and allows communication between the procedures, thereby avoiding tedious storing. Also automatic execution of module occurs when the file is first opened.

```
> restart;
> ode := module()
```

```
> option package;
        export ModuleApply:

  ################ Procedure ModuleApply
  ########################

>    ModuleApply:=proc()
  LbPerGalIn;galPerMinIn;VolOfH2OAtStart;galPerMinOut;
  x0;tmax;so;
  return module()
  export SolveOde, Animate, Next;
>
  use DocumentTools in
    LbPerGalIn:=parse(GetProperty('TA1', 'value'));
    galPerMinIn:=parse(GetProperty('TA2', 'value'));
    VolOfH2OAtStart:=parse(GetProperty('TA3', 'value'));
    galPerMinOut:=parse(GetProperty('TA4', 'value'));
    x0:=parse(GetProperty('TA5', 'value'));
    tmax:=parse(GetProperty('TA6', 'value'));
  end use;
>
> ################ Procedure SolveOde
  ########################


> SolveOde:=proc()
  local ode, ic,x,sol,so;


> ode:=diff(x(t),t)=LbPerGalIn*galPerMinIn -galPerMinOut*x(t)/
  (VolOfH2OAtStart+(galPerMinIn-galPerMinOut)*t);
> ic:=x(0)=x0;
> sol:=dsolve({ode,ic},{x(t)});
> so:=rhs(sol);
  p:=plot(so,t=0..tmax);

  use DocumentTools in
    SetProperty('PL1', 'value', p);
    SetProperty('TA8', 'value', so);
  end use;
>
  so;
  end proc:

  ################ Procedure Animate ########################
> Animate:=proc()
  local a, N, h, i, temp1, temp2, p, two, Salt, b, j, L;

  use DocumentTools in
    Tframe:=parse(GetProperty('TA7', 'value'));
    Salt:=parse(GetProperty('TA8', 'value'));
  end use;
  Salt:=unapply(Salt,t);
> a:=0;b:=tmax;N:=10;h:=(b-a)/N;
> for i from 0  to N do

  if i = 0 then j:=0.001 else j:=i; end if;
  temp1:=plottools[cylinder]([0,0,0], 1, Salt(j*h),
  orientation=[45, 70], scaling=constrained,color=yellow,view=
  [-1..1,-1..1,0..Tframe],axes=boxed):
  temp2:=plots[textplot3d]([-0.7,0.7,Salt(j*h),cat("Salt=",
```

```
convert(Salt(j*h),'string'))],align=RIGHT,color=black,view=
[-1..1,-1..1,0..Tframe],axes=boxed);

p[i]:=plots[display](temp1,temp2):
end do:j:='j';L:=[seq(p[j],j=0..N)];
> p1:=plots[display](seq(p[j],j=0..N),insequence=true,axes=
boxed);

end proc:
>

end module;
end proc;
end module;
Warning, `LbPerGalIn` is implicitly declared local to module
Warning, `galPerMinIn` is implicitly declared local to module
Warning, `VolOfH20AtStart` is implicitly declared local to
module
Warning, `galPerMinOut` is implicitly declared local to
module
Warning, `x0` is implicitly declared local to module
Warning, `tmax` is implicitly declared local to module
Warning, `p` is implicitly declared local to procedure
`SolveOde`
Warning, `Tframe` is implicitly declared local to procedure
`Animate`
Warning, `p1` is implicitly declared local to procedure
`Animate`
```

ode := **module**( ) **option** *package*; **export** *ModuleApply*; **end module** (2.1)

>

>

Output

Lb/gal in = 3 Gal/min in = 4 Vol(gal) of water initially = 300 Gal/min out= 6

Initial amount of salt = 0 t = 0.. 100 Maximum amount of salt from 1st plot = 340

Directions:

Use existing values **or** *modify*. *Click on Solve ode button. View graph below* **and** *enter maximum* in ***Maximum amount of salt*** **from 1*st plot*** *above. Click on* ***Animate*** *button. Then click* ***Play*** *button*

Solve ode

solution(Look below for a prettier solution) =

```
900-6*t+(1/3750)*(-150+t)^3
```

plot of solution =